KNOWLEDGE ENGINEERING FOR TEMPORAL DEPENDENCY NETWORKS AS OPERATIONS PROCEDURES

Kristina E. Fayyad
Monitor & Control Technology Group
M/S 525-3660
Jet Propulsion laboratory
California Institute of Technology
4800 oak Grove Drive
Pasadena, CA 91109
tel. (818) 306-6080
fax (818) 306-6912
email: kristina@isd.jpl.nasa.gov

Topic: Technologies: Information Management, Storage, and Reuse; Monitoring and Adaptive Control

ABSTRACT

DSN Link Monitor & Control operations consist primarily of procedures to configure, calibrate, test, and operate a communications link between an interplanetary spacecraft and its mission control center. The procedures are organized according to two taxonomies: those based on the individual subsystem which form the link (e.g. antenna, receiver, telemetry processor), and those specific to the spacecraft or science experiment. Currently, the operators are responsible for integrating the procedures into an end-to-end series of steps. on-the-job experience provides the necessary background for determining any interdependencies between subsystems and for interpreting mission specific configuration and performance requirements for routine, often-performed operations. Non-routine operations, however, often require the operators to perform different types of operations and to interpret results in ways different than they normally would.

Current research is investigating new ways of specifying operations procedures that incorporate the insight of operations, engineering, and science personnel to make non-routine operations easier. In addition to explicitly defining the interdependencies between subsystems, the end-to-end representation would also provide a means for identifying problems, performing work-arounds, and interpreting monitor data. The difficulty lies in providing a representation which is both human and computer readable.

The approach taken here is to represent procedures as a temporal dependency network (TDN) where high-level procedures are represented as logical nodes of the network. The nodes, or blocks, consist of the directives needed to accomplish the task, temporal constraints, pre- and post- conditions, and local recovery information should the node "fail". The network specifies precedence relationships between nodes, any potential parallelism, and rules for recovering from global faults. Such a representation

can easily be put into a graphical form of boxes and arrows, corresponding to blocks and dependencies, which is readable by the user.

The TDN representation also serves as a starling point for semi-automated operations. An execution module uses the TDN to send and monitor directives. As execution of the TDN progresses, visual changes arc made to the TDN display, e.g. idle, executing, and completed blocks are shown in different colors. in addition, the user can have control over the executing TDN by interacting with the individual blocks in the TDN display.

Recovery capabilities are provided in cases when the TDN needs to be revised during the execution of a pass, for example in the event of a rejected directive, local area network anomaly, or failed device. When such an event occurs, the problem area is identified, the associated blocks arc marked, and a default recovery plan is retrieved which is associated with that event. "l'he operator is notified of the problem and prompted for parameters and/or suggested actions, especially if no recovery plan is available. The resulting information is used to create a "patch" which is spliced into the TDN. If necessary, the execution will be rolled-back to an earlier point in the network,

A prototype system, the Link Monitor and Control Operator Assistant (LMCOA), uses the TDN to represent configuration procedures for an operation. This system was successfully demonstrated by performing precalibration for a Very Long Baseline Interferometry (VLBI) Delta Differential One-Way Ranging (DDOR) pass at the Gold stone 70 Meter Antenna, However, a significant problem in building a TDN is the great amount of knowledge engineering required. Resolving this is particularly important in order to make the usc of TDNs viable operationally.

The following paper describes our efforts of analyzing the knowledge engineering required in order to build a TDN. The goal of these efforts is to gain a better understanding of the functionality and capability of the TDN as a general knowledge representation tool in order to make a decision about the usability of the TDN concept in the automated Monitor & Control domain. We will first formalize the process of building a TDN including the steps of interviewing personnel, reviewing appropriate documentation, identifying device models and attributes, and identifying the sequences of directives within the TDN. We will then identify problems that we had with the process and recommend ways to improve it, The accuracy of the TDN to some degree reflects the accuracy of the information sources accessed during the knowledge engineering process. The usefulness, accuracy, and maintainability of the sources will be reviewed, in addition, we will analyze and formalize the structure of the TDN and report on the knowledge bases associated with the TDN.